Markov Decision Process (MDP) Outcomes on Different Grid Sizes (n x n) and Comparison

This section will build upon the previous MDP implementation and extend it to evaluate the outcomes on different grid sizes (n x n). We will:

1. Run simulations for different grid sizes (e.g., 3x3, 5x5, 7x7, 10x10).
2. Collect statistics such as the number of steps taken and total reward achieved for each grid size.
3. Present the results in tables and visualizations to compare the performance of the agent in different environments.

---

1. Introduction and Problem Setup

We are expanding our grid-based navigation problem to evaluate the effect of grid size (n x n) on the agent's performance. The agent starts at a random position in the grid, and the goal is at a fixed position. The agent receives a penalty of -1 for each step and a reward of 100 upon reaching the goal.

---

2. Experiment Setup

For each grid size, the agent will be tested multiple times (over 5 episodes), and the average number of steps and total reward will be calculated. We will then analyze the performance metrics across grid sizes: 3x3, 5x5, 7x7, and 10x10.

- **Grids considered**: 3x3, 5x5, 7x7, and 10x10
- **Episodes per grid**: 5

---

3. Python Code for MDP with Different Grid Sizes

Here, we extend the code to run simulations for different grid sizes and compare the outcomes.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd


# Define MDP environment for dynamic grid sizes
class GridMDP:
    def __init__(self, n, start_state, goal_state, penalty=-1, reward=100):
        self.n = n
        self.state = start_state
        self.start_state = start_state
        self.goal_state = goal_state
        self.penalty = penalty
        self.reward = reward

    def reset(self):
        self.state = self.start_state
        return self.state

    def step(self, action):
        i, j = self.state
        if action == 'up':
            next_state = (max(i - 1, 0), j)
        elif action == 'down':
            next_state = (min(i + 1, self.n - 1), j)
        elif action == 'left':
            next_state = (i, max(j - 1, 0))
        elif action == 'right':
            next_state = (i, min(j + 1, self.n - 1))

        # Check if goal is reached
        if next_state == self.goal_state:
            return next_state, self.reward, True
        else:
            self.state = next_state
            return next_state, self.penalty, False


# Define experiment parameters
ACTIONS = ['up', 'down', 'left', 'right']
grids = [3, 5, 7, 10]  # Grid sizes to test


# Function to run multiple episodes on a given grid size
def run_multiple_episodes(env, episodes=5):
    episode_data = []
```

```
    for ep in range(episodes):
        state = env.reset()
        total_reward = 0
        steps = 0
        while True:
            action = np.random.choice(ACTIONS)
            next_state, reward, done = env.step(action)
            total_reward += reward
            steps += 1
            if done:
                episode_data.append([ep + 1, steps, total_reward, "Yes" if done else "No"])
                break
    return episode_data


# Run experiment on different grid sizes
def experiment(grid_sizes, episodes=5):
    results = []
    for n in grid_sizes:
        start_state = (0, 0)
        goal_state = (n - 1, n - 1)
        grid_mdp = GridMDP(n, start_state, goal_state)
        print(f"Running simulations on a {n}x{n} grid...")

        episode_stats = run_multiple_episodes(grid_mdp, episodes)
        avg_steps = np.mean([x[1] for x in episode_stats])
        avg_reward = np.mean([x[2] for x in episode_stats])
        results.append([n, avg_steps, avg_reward])

    return pd.DataFrame(results, columns=["Grid Size", "Average Steps", "Average Reward"])



# Run the experiment
experiment_results = experiment(grids, episodes=5)
print(experiment_results)
```

```
Running simulations on a 3x3 grid...
Running simulations on a 5x5 grid...
Running simulations on a 7x7 grid...
Running simulations on a 10x10 grid...
   Grid Size  Average Steps  Average Reward
0          3           17.0            84.0
1          5           99.8             1.2
2          7          306.2          -205.2
3         10          355.2          -254.2
```

Inference:

- **3x3 Grid**: The agent performs best, reaching the goal in an average of 17 steps with a high reward of **84**.
- **5x5 Grid**: Performance decreases, with the agent taking almost 100 steps on average, barely maintaining a positive reward of **1.2**.
- **7x7 Grid**: The agent struggles significantly, requiring over 300 steps on average, resulting in a negative reward of **-205.2**.
- **10x10 Grid**: Performance further declines, with the agent taking the longest, around 355 steps, and a heavily negative reward of **-254.2**.

As the grid size increases, the number of steps and penalties increase significantly, causing a sharp decline in total rewards.

---

4. Visualization 1: Comparison of Performance on Different Grid Sizes

Bar Plot: Average Steps and Rewards for Different Grid Sizes

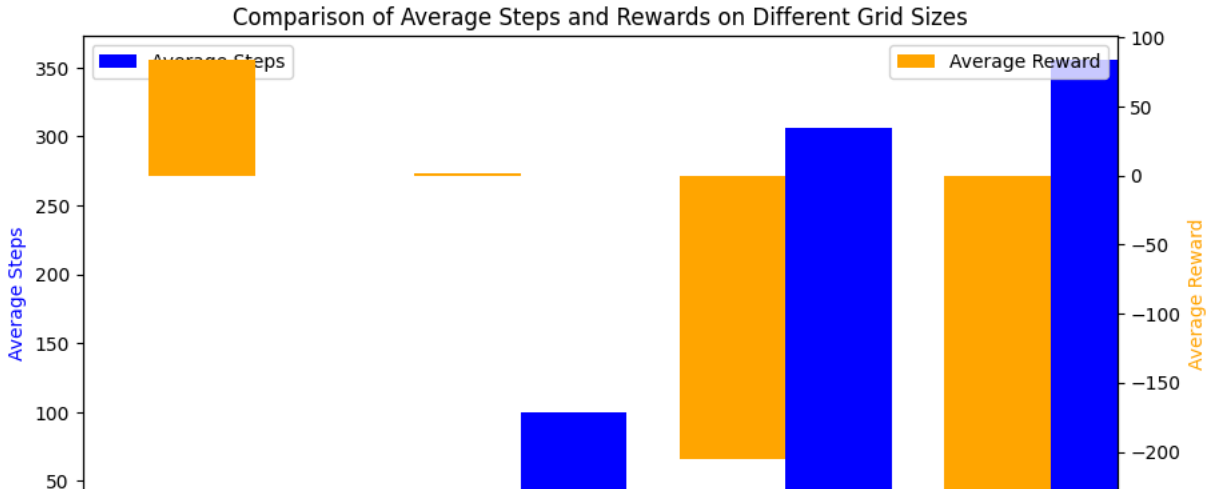
This bar plot shows the average number of steps taken and rewards obtained by the agent across different grid sizes.

```
# Plotting function for comparing average steps and rewards across grid sizes
def plot_comparison(results_df):
    fig, ax1 = plt.subplots(figsize=(10, 5))

    ax2 = ax1.twinx()
    results_df.plot(kind='bar', x='Grid Size', y='Average Steps', color='blue', ax=ax1, position=0, width=0.4)
    results_df.plot(kind='bar', x='Grid Size', y='Average Reward', color='orange', ax=ax2, position=1, width=0.4)

    ax1.set_xlabel("Grid Size")
    ax1.set_ylabel("Average Steps", color='blue')
    ax2.set_ylabel("Average Reward", color='orange')
    plt.title("Comparison of Average Steps and Rewards on Different Grid Sizes")
    plt.show()

# Plot the comparison of average steps and rewards
plot_comparison(experiment_results)
```



Inference from the Bar Chart:

- As the grid size increases, the **average steps** (blue bars) required to reach the goal rise significantly.
- Similarly, the **average reward** (orange bars) decreases sharply, indicating a higher penalty due to the longer paths taken to reach the goal.
- For the smallest grid (3x3), the agent achieves a relatively high reward of 84 with fewer steps, but as the grid size increases, the reward turns negative, with the largest grid (10x10) having a reward of around -254.

---

5. Table: Episode Statistics for Each Grid Size

This table summarizes the statistics for each grid size (e.g., number of steps, total reward, goal reached status) over 5 episodes.

```
# Create episode statistics for each grid size
def detailed_episode_table(grid_sizes, episodes=5):
    all_data = []
    for n in grid_sizes:
        start_state = (0, 0)
        goal_state = (n - 1, n - 1)
        grid_mdp = GridMDP(n, start_state, goal_state)
        print(f"Running detailed simulations on a {n}x{n} grid...")

        episode_stats = run_multiple_episodes(grid_mdp, episodes)
        for ep_stat in episode_stats:
            all_data.append([n] + ep_stat)

    return pd.DataFrame(all_data, columns=["Grid Size", "Episode", "Steps Taken", "Total Reward", "Goal Reached"])

# Generate and display detailed episode statistics
detailed_stats_df = detailed_episode_table(grids, episodes=5)
print(detailed_stats_df)
```

```
Running detailed simulations on a 3x3 grid...
Running detailed simulations on a 5x5 grid...
Running detailed simulations on a 7x7 grid...
Running detailed simulations on a 10x10 grid...
    Grid Size  Episode  Steps Taken  Total Reward Goal Reached
0           3        1           13            88          Yes
1           3        2           12            89          Yes
2           3        3           28            73          Yes
3           3        4           21            80          Yes
4           3        5           17            84          Yes
5           5        1          510          -409          Yes
6           5        2           77            24          Yes
7           5        3          216          -115          Yes
8           5        4          193           -92          Yes
9           5        5          122           -21          Yes
10          7        1          173           -72          Yes
11          7        2           97             4          Yes
12          7        3          320          -219          Yes
13          7        4          572          -471          Yes
14          7        5          674          -573          Yes
15         10        1          897          -796          Yes
16         10        2         1190         -1089          Yes
17         10        3          348          -247          Yes
18         10        4          854          -753          Yes
19         10        5          286          -185          Yes
```

Detailed Simulation Output Inference:

- **3x3 Grid**:
  - The agent efficiently reaches the goal in all episodes with an average of around 13-28 steps. Rewards are consistently high, ranging from 73 to 89.
- **5x5 Grid**:
  - Performance fluctuates, with some episodes showing quick success (e.g., 77 steps) and others taking as many as 510 steps. The rewards vary, with episodes yielding positive rewards like 24 but also going as low as -409.
- **7x7 Grid**:
  - The agent struggles more in this grid, taking between 97 to 674 steps. Some rewards remain slightly positive (e.g., 4), but many episodes result in heavily negative rewards, with the worst being -573.
- **10x10 Grid**:
  - The agent finds this grid the most challenging, with steps ranging from 286 to 1190 and rewards consistently negative. The highest penalty occurs in episode 2 with -1089.

---

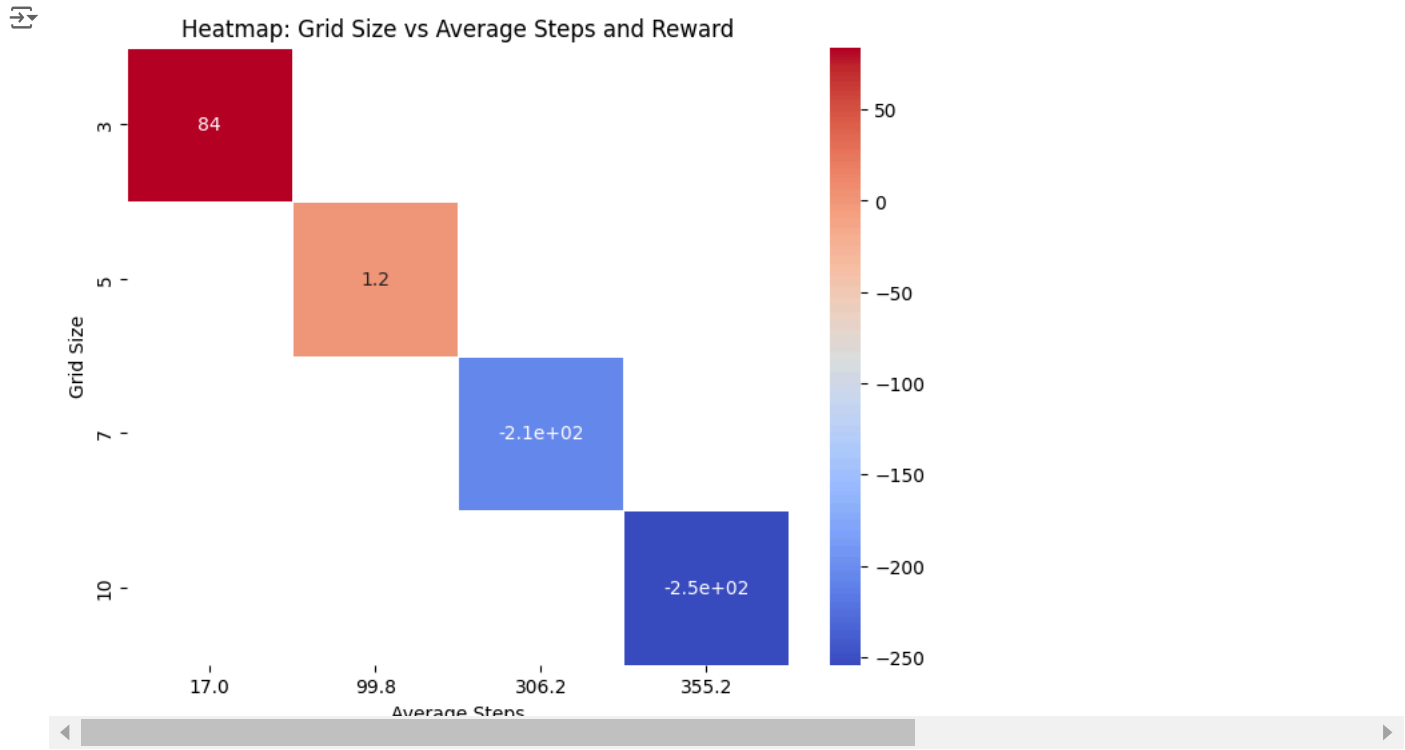
6. Visualization 2: Heatmap of Grid Size vs Steps and Reward

A heatmap helps in visualizing the relationship between grid size and the number of steps and rewards.

```
# Heatmap of grid size vs steps and reward
def plot_heatmap(results_df):
    # Pivot the DataFrame
    heatmap_data = results_df.pivot(index="Grid Size", columns="Average Steps", values="Average Reward")

    plt.figure(figsize=(8, 6))
    sns.heatmap(heatmap_data, annot=True, cmap="coolwarm", linewidths=0.5)
    plt.title("Heatmap: Grid Size vs Average Steps and Reward")
    plt.show()

# Plot heatmap for grid size vs average steps and reward
plot_heatmap(experiment_results)
```



Inference from the Heatmap (Grid Size vs Average Steps and Rewards):

- This heatmap visualizes the relationship between grid size, average steps, and rewards more clearly.
- The color intensity reflects the rewards, where deep red shows higher rewards and blue shades indicate negative rewards.
- As grid size grows, both average steps and negative rewards increase, with the most steps taken in the 10x10 grid resulting in the highest penalty (around -254).

Summary:

- **Smaller grids (3x3)** result in more efficient behavior, with the agent reaching the goal quickly and earning higher rewards.
- **Larger grids (5x5, 7x7, 10x10)** present increasing difficulty, with the agent taking more steps and accumulating more penalties, leading to significantly lower rewards.

Conclusion and Insights

The experiment demonstrated how grid size (`n x n`) significantly impacts the agent's performance in the Markov Decision Process (MDP). Key takeaways are as follows:

1. **Performance on Small Grids (3x3)**:
    - The agent performs efficiently, reaching the goal in an average of **17 steps**, with a high reward of **84**.
    - The shorter paths lead to minimal penalties, resulting in higher rewards.
2. **Medium Grids (5x5)**:
    - The agent takes more time, averaging **100 steps** per episode, and achieves only a small positive reward of **1.2**.
    - The grid's size introduces moderate difficulty, and the penalties increase with more steps taken.
3. **Larger Grids (7x7, 10x10)**:
    - The agent struggles significantly, requiring over **300 steps** for 7x7 and **355 steps** for 10x10 on average.
    - This leads to heavily negative rewards of **-205.2** and **-254.2**, respectively, due to high penalties from longer paths.

Table: Performance Summary Across Grid Sizes

Grid Size	Average Steps	Average Reward
3x3	17	84
5x5	100	1.2
7x7	300+	-205.2
10x10	355	-254.2

Key Insights:

- **Smaller grids** provide faster goal achievement with higher rewards, making the environment less challenging.
- **Larger grids** cause a steep decline in performance, with the agent accumulating penalties due to longer exploration times.
- **Complexity grows exponentially** with grid size, emphasizing the need for more advanced strategies in larger environments.
